REMARKS

Claims 1-37, 40, 42-52, 54, 56, 57, 59-74, 76, 78-80, 82, 87-89, 91, 93, 94, 96-108, 110, and 113-116 are presented for examination, of which Claims 1, 11, 20, 26, 31-33, 51, 52, 65, 73, 74, 78-80, 96, 97, 108, and 113-116 are in independent form. Claims 81, 85, and 109 have been canceled, without prejudice or disclaimer of their subject matter, and will not be mentioned further. Claims 1, 2, 8, 9, 11, 17, 18, 20, 21, 24, 26, 29, 31-33, 44, 45, 51, 52, 62, 65, 73, 74, 78-80, 82, 91, 93, 96, 97, 108, 110, and 113-116 have been amended to define more clearly what Applicants regard as their invention. Favorable reconsideration is requested.

The most recent Office Action, dated November 25, 2003, among other things, rejected Claims 1-37, 40, 42-52, 54, 56, 57, 59-74, 76, 78-80, 82, 87-89, 91, 93, 94, 96-108, 110, and 113-116 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,191,797 (*Politis*).

As noted, Applicants have amended independent Claims 1, 11, 20, 26, 31-33, 51, 52, 65, 73, 74, 78-80, 96, 97, 108, and 113-116 in terms that more clearly define the present invention. Applicants submit that these amended independent claims, together with the remaining claims dependent thereon, are patentably distinct from the cited prior art for at least the following reasons.

The aspect of the present invention set forth, for example, in Claim 33 is a method for optimizing an expression tree. The expression tree represents a compositing expression for compositing an image and comprises a plurality of nodes, where each node of the expression tree represents an object of the image or an operation for combining sub-expressions of the compositing expression. The method includes determining an opacity

region representation for at least one node of the expression tree, the opacity region representation comprising one or more of three predetermined values. Each predetermined value distinctly identifies whether a corresponding region of an object represented by at least one node is an opaque region, a transparent region or a partially transparent region, such that the opacity region representation simultaneously represents each opaque region, transparent region and partially transparent region of the object represented by at least one node. The method also includes optimizing the expression tree by determining an obscurance region representation for at least one node of the expression tree base on an analysis of the opacity region representation associated with at least one node of the expression tree. The obscurance region representation being assigned one or more of a plurality of further predetermined values, where each further predetermined value distinctly identifies whether a corresponding region of at least one object is visible in the image.

In the Advisory Action dated March 24, 2004, the Examiner states that *Politis* teaches the claim limitation of determining an opacity region representation for at least one node of the expression tree (for example, e.g., node 50 of Figures 6-8 wherein the image region representations in hierarchical data structures are given quadtrees).

Accordingly, in a first instance, the Examiner appears to equate the term *opacity region representation* of Claim 33 with the quadtrees of *Politis*. In this connection, at column 14, lines 25-28, *Politis* discusses that in one example an empty quadtree q_0 is created representing the empty foreground region 39 not obscuring the entire image 40. This empty quadtree q_0 is passed to node 51 of the expression tree 50.

In a second instance, in the Advisory Action, the Examiner states that *Politis* teaches the claimed limitation of the opacity region representation being assigned

one or more of three predetermined values, each predetermined value distinctly identifying whether a corresponding sub-region of the image is an opaque region, a transparent region or a partially transparent region (the compositing operations for combining two portions of a single image involves simultaneously identifying one or more of three predetermined values, each predetermined value distinctly identifying whether a corresponding sub-region is an opaque region such as the totally obscured region or opaque sub-image 42 of Figure 6, a transparent region such as the foreground region 39 of Figure 6, or a partially transparent region such as the partially obscured region or the bounding box of text 43 of Figure 6).

Therefore, in the second instance, the Examiner appears to contend that the compositing operations of *Politis* teach each predetermined value distinctly identifying whether a corresponding region of an object represented by the at least one node is an opaque region, a transparent region or a partially transparent region. However, it is unclear in this second instance, whether the Examiner is equating the compositing operations of *Politis* with the term "opacity region representations", since the Examiner does not identify where *Politis* teaches or suggests that compositing operations (i.e., opacity region representations in the second instance) are *assigned one or more of three predetermined values*. Indeed, the Examiner does not identify where *Politis* teaches or suggests that quadtrees (i.e., opacity region representations in the first instance) are *assigned one or more of three predetermined values*.

In any event, Applicants submit that *Politis* does not teach or suggest a determined opacity region representation comprising one or more of three predetermined values, each predetermined value distinctly identifying whether a corresponding region of

an object represented by the at least one node is a partially transparent region, as recited in Claim 33.

At column 5, lines 19-27, *Politis* discusses that at a node of an expression tree represented by an operator, typically a region representation such as a quadtree, is passed to each operand during the process of simplifying the expression tree. Another quadtree corresponding to unobscured portions of graphical elements represented by this node may be returned from the node for possible further processing at other nodes (column 15, lines 26 and 27).

In the description of the second example beginning at column 14, *Politis* discusses a quadtree q_1 that represents the region obscured by the opaque sub-image 42 (column 14, lines 57-60). *Politis* also discusses a quadtree q_0 representing the empty foreground region 39 (column 14, lines 25 and 26).

Politis also discusses that the leaf node 56 of the expression tree receives a quadtree q_2. The quadtree q_2 is compared to the region represented at the leaf node 56 of the expression tree 50. The region represented at the leaf node 56 of the expression tree 50 is the region of the image of Fig. 6 occupied by text 43 (i.e., a bounding box comprising text) (column 15, lines 57-59). *Politis* discusses that the region represented by the quadtree q_2 partly obscures the text 43. In this connection, the Examiner appears to equate the partly obscured region 43 with the term *partially transparent region* of Claim 33.

However, Applicants submit that a partly obscured region is different in meaning to and does not fall within the scope of a *partially transparent region* as claimed in Claim 33. The plain meaning of the term "obscured" is distinctly different from the plain meaning of the term "transparent".

Webster's Revised Unabridged Dictionary©1996, 1998 Micra Inc, defines the term "transparent" as having the property of transmitting rays of light, so that dark bodies can be distinctly seen through; pervious to light; diaphanous; pellucid; as transparent glass; a transparent diamond;– as opposed to opaque.

In addition, as described at page 17, lines 19-29, of the present specification, Fig. 1(b) shows an example of an opacity quadtree 100 describing opacity information of an object 105 shown in Fig. 1(a). The object 105 is a rectangle divided into two halves by a diagonal. One half (i.e., the lower left half) of the object 105 is opaque, while the other, upper right half is partially opaque (i.e., partially transparent). The partitioning of the object 105 is also shown in Fig. 1(a) as square blocks under the object 105. As seen in Fig. 1(b), the opacity quadtree 100 comprises a number of internal (e.g. 101) and leaf (e.g. 103) nodes. Each leaf node (e.g. 103) is assigned a value of 'o', '-' or 'α' which represent the colors black, white and grey respectively, depending on whether a corresponding region in the image space is fully opaque, fully transparent, or partially transparent, respectively.

Therefore, as seen in Fig. 1(b), the second node from the left of level one of the quadtree 100 is grey representing a partially transparent square (i.e., region) of the object 105 at the top right hand corner of the object 105 as seen in Fig. 1(a). As seen in Fig. 1(a) that partially transparent square identified by the second node from the left of level one of the quadtree 100 is partially pervious to light or able to be seen through.

In contrast, *Webster's Revised Unabridged Dictionary*® 1996, 1998 Micra Inc, defines the term "obscure" as to darken; to make dim; to keep in the dark; <u>to hide</u>; to make less visible. Accordingly, taking the plain meanings for the terms "transparent" and "obscured" given above, partially transparent means partially pervious to light, whereas

partly (or partially) obscured means partly hidden. Further, transparency (or being transparent) is a property of an object, whereas, something being obscured (or hidden) is the result of an action by another object. This is consistent with manner in which the term "obscure" is used in *Politis*. For example, at column 2, lines 54-60, *Politis* states that another way in which only a portion of an element may have an effect is when the portion (i.e., the portion of the element) is obscured by another element.

Accordingly, Applicants submits that *Politis* does not teach or suggest the identification of a *partially transparent region*. At most, *Politis* only discusses a partly obscured region (i.e, the text region 43).

In any event *Politis* actually teaches away from determining a quadtree or region representation representing the partly obscured text region 43. *Politis* discusses that an empty quadtree is preferably returned rather than expend processing time to achieve a quadtree representation of the region of text 43 (column 16, line 5-10), since the element text "hello" 43 does not substantially obscure other graphical elements.

Further, Applicants submit that *Politis* does not teach or suggest determining an opacity region representation such that the opacity region representation simultaneously represents each opaque region, transparent region and partially transparent region of the object represented by said at least one node of an expression tree, as recited in Claim 33.

In contrast, as discussed above, three quadtrees are disclosed in the second example of *Politis* (i.e., quadtree q_1 representing the opaque sub-image 42, quadtree q_0 representing the empty foreground region 39 and an empty quadtree q_3 for regions of the image occupied by text 43). *Politis* does not teach or suggest a quadtree or region

representation simultaneously representing each of the opaque sub-image 42, the empty foreground region 39 and the partly obscured text 43. Even if *Politis* did teach such a quadtree, the quadtree would not simultaneously identify each opaque region, transparent region and partially transparent region of an object represented by a node, since *Politis* only teaches identifying partly obscured (i.e., partly hidden) regions such as the text region 43.

At column 3, line 62, to column 4, line 63, *Politis* discusses compositing operations, as shown in Table 1, for combining two portions of an image. The symbols for the compositing operations are described as Dc representing a premultiplied destination or resultant color, Do representing a destination or resultant alpha (α) channel value, Ac representing a premultiplied pixel color of a first portion of a first source A, Ao representing an α value corresponding to the pixel having the color Ac, Bc representing a premultiplied pixel color value of a portion of an image of a second source B, and Bo representing the α channel value of the pixel corresponding to Bc of the source B.

The Examiner states that the compositing operations for combining two portions of a single image involves simultaneously identifying one or more of three predetermined values. Even if, to combine two portions of an image, a compositing operation has to identify one or more of three predetermined values. Applicants submit that *Politis* in general and particularly at column 3, line 62, to column 4, line 63, does not teach or suggest a compositing operation comprising one or more of three predetermined values, each predetermined value distinctly identifying whether a corresponding region of an object represented by at least one node is an opaque region, a transparent region or a partially transparent region such that the compositing operation simultaneously identifies

each opaque region, transparent region and partially transparent region of the object represented by at least one node, as recited in Claim 33.

Accordingly, Applicants submit that Claim 33 is clearly patentable over *Politis*. Claims 11, 26, 32, 52, 65, 74, 78, 97, 114 and 116 include features similar to those discussed above in connection with Claim 33. Accordingly, Claims 11, 26, 32, 52, 65, 74, 78, 97, 114 and 116 are believed to be patentable for reasons substantially similar as those discussed above in connection with Claim 33.

The aspect of the present invention set forth in Claim 1 is a method of creating an image. The image being formed by rendering at least a plurality of graphical objects to be composited according to an expression tree representing a compositing expression for the image. The expression tree includes a plurality of nodes each representing an object of the image or an operation for combining sub-expressions of the compositing expression. The method includes determining an opacity region representation for at least one node of the expression tree. The opacity region representation comprising one or more of three predetermined values, each predetermined value distinctly identifies whether a corresponding region of an object represented by at least one node is an opaque region, a transparent region or a partially transparent region, such that the opacity region representation simultaneously represents each opaque region, transparent region and partially transparent region of the object represented by at least one node. A union of each opacity region representation for the expression tree includes at least one of each of the three predetermined values. The method also includes determining an obscurance region representation for at least one node based on an analysis of the opacity region representation associated with at least one node of the expression tree. The

obscurance region representation being assigned one or more of a plurality of further predetermined values, each further predetermined value distinctly identifying whether a corresponding region of the object represented by at least one node is visible in the image. The method further includes partitioning the object into a plurality of regions, overlaying the obscurance region representation on the partitioned object such that the partitioned object is substantially encompassed within the obscurance region representation, traversing the overlaid obscurance region representation to identify any of the plurality of regions of the partitioned object which include at least a portion of the visible region, and creating the image by rendering the identified regions.

As discussed above, the present specification describes at page 17, lines 19-29, an example of an opacity quadtree 100. The opacity quadtree 100 comprises a number of internal (e.g. 101) and leaf (e.g. 103) nodes. Each leaf node (e.g. 103) is assigned a value of 'o', '-' or 'α' which represent the colors black, white and grey respectively, depending on whether a corresponding region in the image space is fully opaque, fully transparent, or partially transparent, respectively.

Accordingly, the quadtrees of the present invention are tripartite quadtrees, where each leaf node (e.g. 103) is assigned one of three predetermined values. One characteristic of such a quadtree or region representation, as claimed in Claim 1, is that a union of each quadtree or region representation for an expression tree includes at least one of each of the three predetermined values.

In contrast, as disclosed at column 3, lines 40-50, of *Politis*, image region representations are hierarchical data structures suitable for representing a region or portion of an image and typically used in image processing. *Politis* further discusses that one such

image region representation is known to those skilled in the art as quadtrees. *Politis* discusses that an image region representation is referred to as a quadtree. *Politis* further discusses that the creation of a quadtree representing a region of an image requires the sub-division of the region into a plurality of cells, each cell being a portion of the region, and each cell represented by a node of the quadtree. Further, at column 7, lines 20-64, *Politis* discusses that each node of the expression tree receives from its parent node a region representation of one or more areas of the image. The region representation is compared to the region represented at the node to determine if the region represented by that node is obscured.

At Figs. 6-8, *Politis* depicts an image expression tree (50), which represents a rendered image (40) composed of simple graphic objects. An empty quadtree q_0 is created representing the empty foreground region (39) not obscuring the entire image (40), as described at column 14, lines 25-28, of *Politis*. This empty quadtree q_0 is passed to a first node (51) (or root node) of the expression tree (50). As described at column 14, lines 48-54, a third node (53) is a leaf node representing the sub-image (42). This third node (53) receives the quadtree q_0 passed down from a second node (52) and compares the region of the node (53) with the region represented by the quadtree q_0 to determine if the region represented by node (53) is obscured by the quadtree q_0.

In the second example of *Politis*, the quadtree q_0 is empty and therefore the node (53) is not obscured. However, the image "A" is a graphical element that can potentially obscure other graphical elements. Hence, a quadtree q_1 that represents the region obscured by the image is created, and passed back to the second node (52) since no further left branches are available at the third node (53).

Further, as discussed at column 15, lines 53, to column 16, lines 20, of *Politis* the left descendent of a fifth node (55) is a leaf node (56) representing the region of the image (40) illustrating the text (43). The leaf node (56) receives a quadtree q_2 passed down from the fifth node (55) and is compared to the region represented at the leaf node (56) (typically, the region of the image (40) occupied by the text (43) is a bounding box comprising text) to determine if the region represented by quadtree q_2 obscures the region represented at leaf node (56). The region represented by the quadtree q_2 (the region occupied by circle (44)) partly obscures text (43). Hence, the text (43) is clipped or tagged for clipping at a later stage. The text (43) is clipped by applying a clipping operator, wherein the clipping operation constructs a "clip" path from the quadtree q_2 and clips or cuts the text (43) to this path. At this point, a new quadtree representing the region of the image occupied by the text is created and returned to the fifth node (55). However, if a graphical element is too small to substantially obscure other graphical elements of the image (e.g. the graphical element text "hello" (43) does not substantially obscure other graphical elements even though the bounding box of text (43) represents a substantial region), an empty quadtree is preferably returned rather than expend processing time to achieve a quadtree representation of the region of text (43). Hence, the creation of a new quadtree q_3 for regions of the image occupied by text (43) is chosen as an empty quadtree. While a quadtree representation for text (43) can be created, the cost in the performance speed of the process out-weighs the time it takes to render text. Hence, the empty quadtree q_3 is created and passed back to the fifth node 55.

Accordingly, the quadtrees of *Politis* are binary quadtrees. In particular, <u>empty</u> quadtrees are used to represent portions of an image <u>not obscuring</u>, while, if a region

obscures the image, then the obscuring portion of the image is represented by a quadtree. As such, a union of such quadtrees for an expression tree would only contain at least one of each of "two" values. For example, at least one value representing an obscured region and at least one value (i.e., empty) representing a region not obscuring.

For at least the reasons given above, Applicants submit that *Politis* does not teach or suggest that a union of each opacity region representation for the expression tree includes at least one of each of the three predetermined values, as recited in Claim 1.

For reasons similar to those discussed above in connection with Claim 33 and for those reasons discussed directly above, Applicants submit that Claim 1 is clearly patentable over *Politis*. Furthermore, Claims 20 and 31 are apparatus and computer program claims respectively corresponding to method Claim 1, and are believed to be patentable for at least the same reasons as discussed above in connection with Claim 1.

The aspect of the present invention set forth in Claim 51 is a method for optimizing an expression tree. The expression tree represents a compositing expression for compositing an image and comprises a plurality of nodes, each node of the expression tree represents an object of the image or an operation for combining sub-expressions of the compositing expression. The method includes determining an opacity quadtree for at least one node of the expression tree, each leaf node of the opacity quadtree being assigned one of three predetermined values, each predetermined value distinctly identifying whether a corresponding sub-region is an opaque region, a transparent region or a partially transparent region. The method also includes optimizing the expression tree by determining an obscurance quadtree for at least one node of the expression tree using the opacity quadtree associated with at least one node of the expression tree. The obscurance

quadtree being assigned one of a plurality of further predetermined values, each further predetermined value distinctly identifying whether a corresponding sub-region is visible in the image.

Applicants submit that *Politis* in general and particularly Figs. 6-8 fails to teach or suggest determining an opacity quadtree for at least one node of the expression tree, each leaf node of the opacity quadtree being assigned one of three predetermined values, each predetermined value distinctly identifying whether a corresponding sub-region of the image is an opaque region, a transparent region or a partially transparent region, as recited in Claim 51.

For example, as described at page 17, lines 10-18, of the present specification, for obscurance analysis, the homogeneous regions of the image are of interest. That is, those regions corresponding to parts of objects that are of different levels of opaqueness (i.e., fully opaque, fully transparent or partially transparent regions). These regions are needed in order to take full advantage of possible optimization opportunities that exist in the presence of the various operations (i.e., OVER, IN, OUT, ATOP, etc.). Operators such as OVER require the fully opaque regions to determine where one object follows another, whereas operators such as IN and OUT require the fully transparent regions to determine how an object clips out parts of other objects.

Further, as disclosed at page 17, lines 25-28, of the present specification, each leaf node of a quadtree is assigned a value representing the colors black, white and grey respectively, depending on whether a corresponding region in the image space is fully opaque, fully transparent or partially transparent, respectively. This particular feature provides those regions corresponding to parts of objects that are of different levels of

opaqueness (i.e., fully opaque, fully transparent or partially transparent regions) in order to take full advantage of possible optimization opportunities that exist in the presence of the various operations (i.e., OVER, IN, OUT, ATOP, etc.).

In contrast, as discussed above, the quadtrees of *Politis* are used to represent a region or portion of an image. In particular, empty quadtrees are used to represent portions of an image not obscuring, while, if a region obscures the image, then the obscuring portion of the image is represented by a quadtree. However, *Politis* fails to teach or suggest determining an opacity quadtree for at least one node of the expression tree, each leaf node of the opacity quadtree being assigned one of three predetermined values, as recited in Claim 51. That is, the opacity quadtree of Claim 51 is a tripartite quadtree.

In making the rejection in the Advisory Action, the Examiner states that *Politis* teaches optimizing the expression tree by determining an obscurance region representation for at least the node of the expression tree (e.g. determining an obscurance region representation for the node 50 of Figures 6-8) based on an analysis of the opacity region representation associated with the node of the expression tree (either the opaque sub-image 42 of Figure. 6 or the bounding box of text 43 of Figure 6 associated with the node 50 of Figure 6, because the resolution of the region is represented by the quadtree), the obscurance region representation being assigned one or more of a plurality of further predetermined values, each further predetermined value distinctly identifying whether a corresponding sub-region is visible in the image.

Applicant submits that *Politis* in general and particularly Figs. 6-8 fails to teach or suggest optimizing the expression tree by determining an obscurance quadtree for at least the one node of the expression tree using the opacity quadtree associated with at

least one node of the expression tree, each leaf node of the obscurance quadtree being assigned one or more of a plurality of further predetermined values, each further predetermined value distinctly identifying whether a corresponding sub-region of the image is visible in the image, as recited in Claim 51.

As described at page 20, lines 23-29, of the present specification, obscurance analysis requires knowledge of the opacity information of each object, in the form of opacity quadtrees, so that regions where objects have been hidden or clipped out by other regions can be identified. These obscured regions are generally irregular in shape and can also be represented using quadtrees, referred to as obscurance quadtrees. Unlike opacity quadtrees, obscurance quadtrees preferably contain only two distinct node values instead of three. The two distinct node values being '1' where the object is hidden, and '0' where the object is visible. Obscurance quadtrees are computed from the opacity quadtrees, as described at page 21, lines 12-22, of the present specification. An obscurance quadtree represents the union of all obscured regions represented by a corresponding leaf node. As each node in a compositing tree inherits the obscured regions of the nodes parent node, the obscurance quadtrees are propagated in a downwards tree traversal. The process concludes with a final obscurance quadtree which can be used to limit the amount of processing required to render the graphics object corresponding to the particular leaf node.

In contrast, as described above, the quadtrees of *Politis* are merely region representations of one or more areas of an image that are passed down to a node from a parent node and compared to the region represented at the node (column 7, lines 25-28). Applicants have found nothing in *Politis* that would teach or suggest an obscurance quadtree being determined for at least the one node of the expression tree <u>based on an</u>

analysis of the opacity quadtree associated with at least one node of the expression tree, as recited in Claim 51.

For at least the above reasons, Applicants submit that Claim 51 is clearly patentable over *Politis*. Claims 73, 79, 80, 96, 108, 113 and 115 include features similar to those discussed above in connection with Claim 51. Accordingly, Claims 73, 79, 80, 96, 108, 113 and 115 are believed to be patentable for reasons substantially similar as those discussed above in connection with Claim 51.

The other rejected claims in this application depend from one or another of the independent claims discussed above, and, therefore, are submitted to be patentable for at least the same reasons. Since each dependent claim is also deemed to define an additional aspect of the invention, individual reconsideration of the patentability of each claim on its own merits is respectfully requested.

In view of the foregoing amendments and remarks, Applicants respectfully request favorable reconsideration and early passage to issue of the present application.

Applicants' undersigned attorney may be reached in our New York office by telephone at (212) 218-2100. All correspondence should continue to be directed to our below listed address.

Respectfully submitted,

Ronald A. Clayton
Attorney for Applicants

Registration No. 26,718

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3801
Facsimile: (212) 218-2200

NY_MAIN 436536